33
150pts.

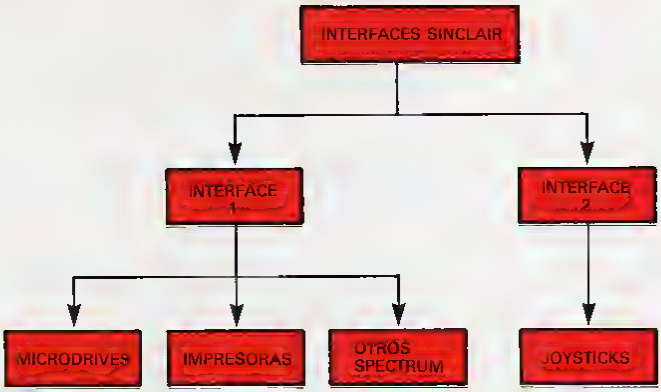
RUN

Enciclopedia Práctica del Spectrum

Nueva Lente/Ingelek

EL INTERFACE 1

DOS son los *interfaces* creados por *Sinclair Research Ltd* para potenciar el desarrollo de la versión básica de Spectrum. El primero de ellos, denominado ZX INTERFACE 1, encaminado a manejar hasta 8 unidades MICRODRIVE en línea, controlar la red de área local (interconexión de hasta 64 ordenadores) y acceso a impresoras y otros periféricos adscritos al estándar de serie RS232C, el segundo, denominado ZX INTERFACE 2, destinado a proporcionar las máximas prestaciones en cuanto al manejo de juegos, permitiendo el uso de hasta dos *joysticks*



Dos son los interfaces creados por Sinclair Research Ltd. para propiciar el desarrollo del Spectrum: el ZX INTERFACE 1 y el ZX INTERFACE 2.

CANALES Y CORRIENTES

Ya hemos visto al hablar de la unidad MICRODRIVE la forma en que se abren, cierran y actualizan los ficheros de datos. Estas funciones se realizan siempre apoyándonos en la asociación de los números de corriente y canal especificados como parámetros de la sentencia OPEN #.

Tanto el número de corrientes como el de canales es limitado. Pueden definirse hasta 16 corrientes (o números de fichero) numeradas de 0 a 15 y asignadas a cualquiera de los canales disponibles en el ordenador.

Para hacernos una idea más concreta de lo que debemos entender por canales y corrientes, podemos decir que los canales son como carreteras que unen determinadas partes de nuestro ordenador y las corrientes vehículos que hacemos circular por ellas.

Por ello, puede darse la circunstancia concreta de que en un canal no concurra ninguna corriente, de la misma forma que en una determinada puede coincidir más de un canal simultáneamente.

En una primera clasificación podemos dividir los canales en tres tipos: los de entrada, los de salida y los mixtos. Como ejemplo del primer grupo tenemos el teclado del ordenador, del segundo la pantalla y la ZX PRINTER, y del tercero las unidades MICRODRIVE, la red de área local y el RS232C.

Está claro que la unidad de entrada de datos por excelencia es el teclado. Esto es debido a que se trata de un periférico especializado en esta labor, hacia el cual no puede efectuarse ninguna salida. Del mismo modo, tanto la pantalla como la

Los canales son como carreteras que unen determinadas partes de nuestro ordenador, y las corrientes los vehículos que circulan por ellas

NUEVA CORRIENTE

OPEN # 4, "S"

LA REVOLUCION DE LA TECNICA

Podemos dividir los canales en tres tipos entrada (teclado), salida (pantalla, impresora) y mixto (Microdrive, red local, RS 232 C)

impresora son unidades de salida por naturaleza, puesto que ninguna de ellas canaliza una entrada de datos sino todo lo contrario.

Por último, existe un tercer grupo de periféricos más versátil, cuya utilización se adapta exactamente a nuestras necesidades. Este es el caso de las unidades MICRODRIVE en las cuales puede almacenarse o leerse información a nuestro gusto, o la red de área local a través de la cual comunicamos nuestro ordenador con otros.

Los 7 canales del Spectrum son:

Podemos crear corrientes asignadas a la pantalla e impresora con **OPEN #**, *definiendo el número de corriente y especificando el canal*

"k" para el teclado *(keyboard)*
"s" para la pantalla *(Screen)*
"p" para la impresora *(Printer)*
"m" para las unidades MICRODRIVE *(Microdrive)*
"n" para la red de área local *(Net)*
"t" y "b" para el RS232 *(Text & Bytes)*

Los 3 primeros canales, "k", "s" y "p", están preestablecidos, siendo necesario en la sintaxis de la sentencia **OPEN #** el empleo de la coma (,) como separador. En los demás, como ya hemos visto al hablar de la sentencia **OPEN #** aplicada a los ficheros en MICRODRIVE, puede emplearse el punto y coma (;). Las siguientes expresiones son, por lo tanto, equivalentes:

```
OPEN #c,"m",n,"nombre-fichero"
OPEN #c;"m";n;"nombre-fichero"
```

Recordemos también que algunos números de corriente son asignados automáticamente por el ordenador al ponerlo en funcionamiento. De este modo lo son:

#0 y #1 a la salida a la línea de estado (línea 24) y entrada desde el teclado.
#2 a la pantalla en general (líneas 0 a 21).
#3 a la impresora.

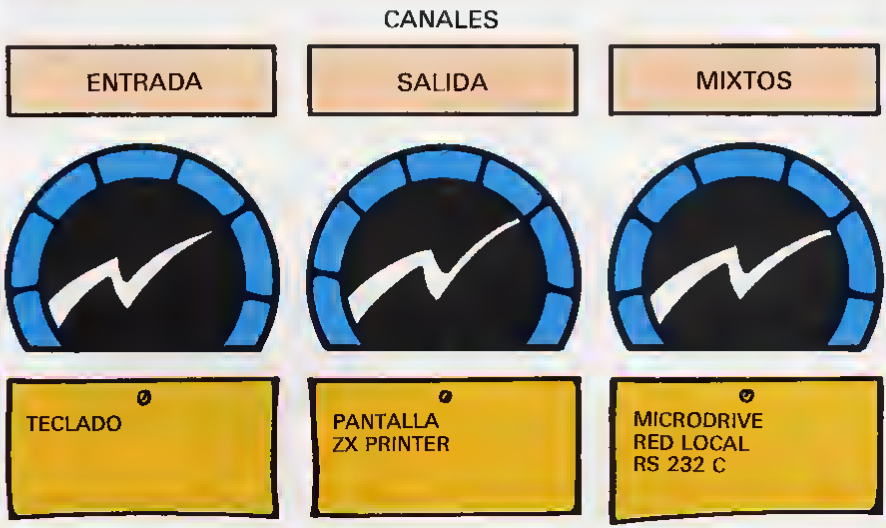
Así, podemos decir que **PRINT** es una notación abreviada de **PRINT # 2**, del mismo modo que **LPRINT** lo es de **PRINT #3.** De hecho, las sentencias **PRINT** y **LPRINT** son equivalentes con la única diferencia de tener asignadas por defecto las corrientes 2 y 3 en los canales "s" y "p", respectivamente.

¡!

El ZX INTERFACE 1 permite manejar hasta 8 unidades MICRO DRIVE controlar la red de área local y el acceso a impresoras y otros periféricos RS232

*

Pueden definirse hasta 16 corrientes (o numeros de fichero) numeradas de 0 a 15 y asignadas a cualquiera de los canales disponibles en el ordenador



CANALES SPECTRUM

Los siete canales del Spectrum son: k (teclado), s (pantalla), p (impresora), m (microdrive), n (red de área local), t (texto RS 232 C) y b (bytes RS 232 C).

Otra posibilidad es la de crear nuevas corrientes asignadas a pantalla o impresora. Para ello basta con definir el canal en concreto. Las siguientes líneas pueden servir de ejemplo:

```
OPEN #4,"s"
OPEN #5,"p"
```

En lo sucesivo, las sentencias **PRINT #4** y **PRINT #5** tendrán salida a pantalla e impresora, respectivamente.

LA SENTENCIA MOVE

Ahora que ya conocemos un poco más a fondo el funcionamiento de los canales y las corrientes, podemos hacer uso de una sentencia BASIC que nos permite trasladar ficheros de datos de unas corrientes a otras, y obtener copias de seguridad de los mismos, la sentencia **MOVE**. Su formato general es:

```
MOVE #c1 TO#C2
```

Donde el parámetro c1 especifica el número de la corriente fuente y c2 el de la corriente destino. Una de las aplicaciones más frecuentes de esta sentencia es la exploración por pantalla del contenido de ficheros en cartucho, así como la duplicación de éstos. De esta forma podemos examinar por pantalla el fichero de agenda telefónica obtenido en el capítulo anterior:

```
MOVE "m";1;"TELEFONOS" TO #2
```

Para realizar una copia de seguridad de éste podemos escribir:

```
MOVE "m";1;"TELEFONOS" TO "m";1;"TELEFONOS2"
```

Si disponemos de dos unidades MICRODRIVE, podemos copiar el contenido del fichero en la unidad 1 a la unidad 2 de la forma:

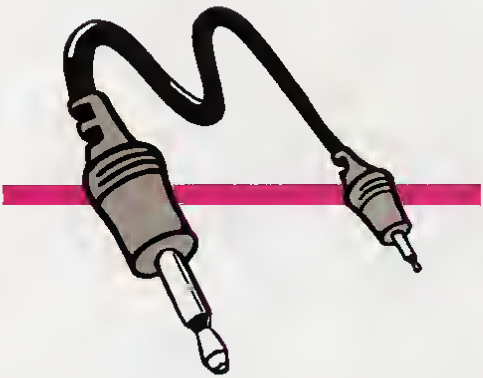
La sentencia **MOVE** *te permite trasladar ficheros de datos de unas corrientes a otras y obtener copias de seguridad*

¡!

Los canales son como carreteras que unen determinadas partes de nuestro ordenador y las corrientes vehículos que hacemos circular por ellas.

*

Podemos dividir los canales en tres tipos: los de entrada, los de salida y los mixtos.



Con el empleo del cable suministrado con cada ZX INTERFACE 1 se puede establecer la conexion de hasta 64 Spectrums

Cada ordenador debe tener su conector derecho ocupado por la clavija del cable de entrada proviniente del ordenador anterior, y el de salida del propio ordenador (izquierdo) comunicado con el siguiente ordenador en la línea. De esta forma, el primero de la línea debe tener ocupado únicamente uno de los conectores, del mismo modo que el último, mientras que el resto de los ordenadores deberán tener los dos ocupados.
Superada la fase de conexión, debe asignarse un número a cada estación de trabajo para poder ser reconocida en el envio y recepción de informacion. Este cometido lo realiza la sentencia **FORMAT**:

FORMAT "n";e

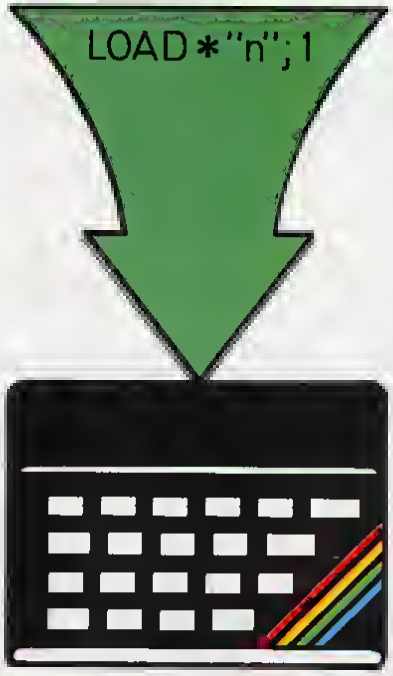
Donde e representa el número de la estación. Como excepción, si existen solamente 2 ordenadores en una red, ambos pueden utilizar el mismo número de estación, que por defecto se corresponderá con el número 1, no siendo necesario efectuar los **FORMAT.**

MOVE "m";1;"TELEFONOS" TO "m",2;"TELEFONOS"

Por último, para conseguir una salida simultánea de un catálogo de cartucho, o los resultados de un programa a pantalla e impresora, podemos escribir

MOVE #2 TO #3

La transferencia de programas entre ordenadores dentro de la red se realiza de forma similar al almacenamiento de estos en cartucho



LA RED DE AREA LOCAL

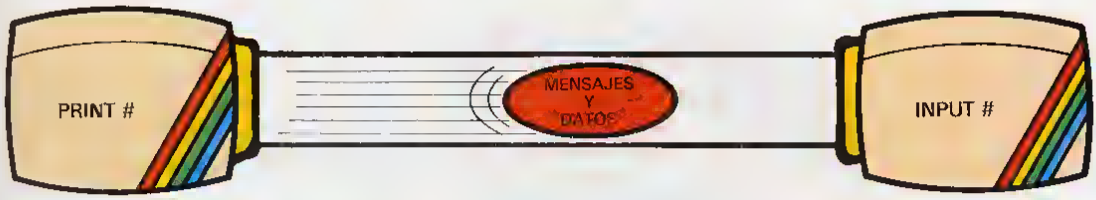
El establecimiento de la red de área local es, junto con la utilización de unidades MICRODRIVE y el acceso a periféricos e impresoras profesionales del tipo «serie», una de las posibilidades más importantes que brinda el ZX INTERFACE 1.
Con el empleo del cable suministrado con cada ZX INTERFACE 1 puede establecerse la conexión de hasta 64 Spectrums. Para ello basta con conectar el cable a uno de los conectores (similares a los de MIC y EAR de casete) situados en la parte posterior central del ZX INTERFACE 1. Esta interconexión de ordenadores debe realizarse de forma que cada estación de trabajo quede comunicada con la siguiente

¡!

Los 7 canales del Spectrum son **"k"** para el teclado, **"s"** para la pantalla, **"p"** para la impresora, **"m"** para las unidades MICRODRIVE, **"n"** para la red de area local y **"t"** y **"b"** para el RS232

*

Los canales **"k"**, **"s"** y **"p"**, están preestablecidos siendo necesario en la sintaxis de la sentencia **OPEN #** el empleo de la coma (,) como separador



La red permite igualmente la transferencia de mensajes y datos de un ordenador a otro, sirviéndonos de las sentencias **PRINT #** *e* **INPUT #.**

TRANSFERENCIA DE PROGRAMAS Y DATOS

La transferencia de programas entre ordenadores dentro de la red se realiza de forma similar al almacenamiento de éstos en cartucho.

Supongamos que disponemos de dos estaciones con los números 1 y 2. Pues bien, para transferir un programa desde la estación 1 a la 2, la primera debe cargar el programa en memoria y acto seguido teclear:

SAVE *"n";2

Con ello la estación 1 queda a la espera de realizar el envío hacia la estación 2, cosa ésta apreciable por pasar el contorno de la pantalla al color negro. Seguidamente, desde la estación 2 debe teclearse:

LOAD *"n";1

Si bien **SAVE *** es la única forma de enviar programas de un ordenador a otro, existen 3 formas diferentes de recibirlos desde la otra estación: **LOAD ***, **VERIFY *** y **MERGE ***. Su utilización depende de nuestras necesidades. Si, por ejemplo, deseáramos verificar el envío del programa anterior, deberemos repetir la operación **SAVE *** en la estación 1 y teclear en la estación 2:

VERIFY *"n";1

La red permite igualmente la transferencia de mensajes y datos de un ordenador a otro, sirviéndonos de las sentencias **PRINT #** e **INPUT #**, en la estación emisora y receptora respectivamente. Debemos tener presente que, de forma similar a como sucede con los ficheros de datos, la información enviada por **PRINT #** es almacenada en un *buffer* intermedio de 256 bytes. Es decir, la información no se manda a la estación de destino hasta completarse este espacio. Por ello, para forzar el envío de mensajes debemos acompañar a la sentencia **PRINT #** de un **CLOSE**. En definitiva, estamos obligados a la apertura del fichero, el envío del mensaje y el cierre del mismo, por cada nuevo envío.

La estación receptora de los datos puede recibir éstos de dos formas: a través de **INPUT #** o de **INKEY$ #**. En el primer caso se recibirá un mensaje completo, mientras que en el segundo la recepción se efectuará carácter a carácter.

Al iniciar el envío de información el contorno de la pantalla pasa a negro, permaneciendo en este estado hasta que pueda realizarse la recepción desde la estación de destino. Así, al concluir el envío recibiremos el mensaje de **O.K.**, volviendo el contorno a su color habitual.

Existe además otra forma de envío denominada

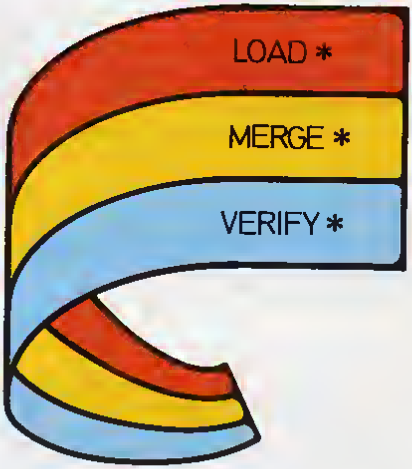
Una de las aplicaciones mas frecuentes de **MOVE** *es la exploración por pantalla del contenido de ficheros en cartucho.*

¡!

El canal **"t"** supone caracteres de 7 bits, por lo que si alguno tuviera 8 este último sería ignorado.

*

El canal **"b"** utiliza los codigos completos de 8 bits, permitiendo el envío de los codigos de control de las impresoras, y la transmisión de programas o estructuras complejas de información hacia o desde otro periférico de este tipo de *interface*.



SAVE es la unica forma de enviar programas de un ordenador a otro, pero existen tres formas de recibirlos desde otra estación* **LOAD***, **VERIFY*** *y* **MERGE***.

«difusión» que permite la recepción simultánea desde todas las estaciones de la red especificada como **"n"; 0**. Esta forma especial es inmediata, no precisando del cierre de ficheros y sin esperar a que las estaciones receptoras estén preparadas para recibir la información

Si, por ejemplo, deseamos enviar un programa a toda la red basta con escribir en la estación emisora y receptoras, respectivamente

SAVE *"n",0
LOAO *"n",0

Eso sí, debemos poner especial cuidado en colocar primero todas las estaciones en espera con su **LOAD *** antes de realizar el envío desde la estación emisora puesto que, en este único caso, no se recibe el mensaje de *O K* al envío

EL INTERFACE RS232

El interface RS232 permite enviar y recibir datos al conectarlo a cualquier periférico de serie compatible, como una impresora, otro ordenador, o un modem de comunicaciones

La conexión de cualquier periférico requiere el uso de una clavija del tipo «D» de 9 vías, alojada en el conector situado al efecto en el panel posterior del ZX INTERFACE 1, así como un cable de unión y una clavija de entrada adecuada al periférico que deseamos enganchar

Una vez cumplidos los requisitos *hardware* es necesario hacer una serie de puntualizaciones sobre cómo el Spectrum gestiona las transmisiones con el RS232

— La velocidad de transmisión en baudios (bits por segundo) debe ajustarse a alguna de las siguientes: 50, 110, 300, 600, 1200, 2400, 4800,

A través del canal "b" es posible la comunicación de dos Spectrum por la línea telefónica con el empleo de un modem.

¡!

Algunos numeros de corriente son asignados automaticamente por el ordenador al ponerlo en funcionamiento **# 0** y **# 1** a la salida a la linea de estado y entrada desde el teclado **# 2** a la pantalla en general, y **# 3** a la impresora

*

MOVE es una sentencia BASIC que permite trasladar ficheros de datos de unas corrientes a otras

*

Con el empleo del cable suministrado con cada INTERFACE 1 puede establecerse la conexion de hasta 64 Spectrums

*

Superada la fase de conexion debe asignarse un numero a cada estacion de trabajo para poder ser reconocida en el envio y recepcion de información

La velocidad de transmisión en baudios entre el interface RS 232 C y los periféricos debe ajustarse a alguna de las siguientes: 50, 110, 300, 600, 1200, 2400, 4800, 9600 ó 19200

9600 ó 19200. En cualquier caso, deberemos seleccionar la más alta de las posibles en el periférico.

— En las impresoras, el avance de línea automático debe estar desactivado. Esto se debe a que el Spectrum envía por cada final de línea los códigos 13 de retorno de carro y 10 de avance de línea.

— La paridad debe estar desactivada.

— El Spectrum emite un sólo bit de parada.

Las transmisiones vía RS232 se realizan a través de dos canales diferentes: el "t" para el envío de datos y el "b" para los códigos de control.

El canal "t" supone caracteres de 7 bits, por lo que si alguno tuviera 8 éste último sería ignorado. La interpretación que de los códigos ASCII enviados por este canal hace el Spectrum, es la siguiente:

CODIGOS	RESULTADOS
0- 31	Estos códigos no se envían a excepción del retorno de carro que lo es como un carácter 13 y un carácter 10.
32-127	Envío normal del carácter
128-164	Estos códigos no se envían.
165-255	Se traducen a palabras formadas por los caracteres ASCII.

El canal "b" utiliza los códigos completos de 8 bits, permitiendo el envío de los códigos de control de las impresoras, y la transmisión de programas o estructuras complejas de información hacia o desde otro periférico de este tipo de *interface*.

EL CANAL "t"

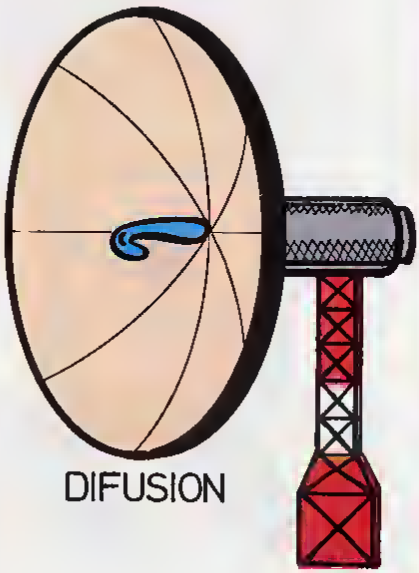
La primera operación a realizar antes de emplear el canal "t" es determinar la velocidad en baudios a que tendrá lugar la transmisión, seleccionando la adecuada al periférico de entre las posibles en el Spectrum. La sentencia que realiza esto adopta el formato:

FORMAT "t";v

Donde v corresponde a la velocidad.

A partir de aquí puede asignarse una corriente con la sentencia **OPEN**)c,"t", y efectuar tanto operaciones de lectura por medio de **INPUT**)c o INKEY$)c, como de escritura con **PRINT**)c, don-

Existe una forma de envío de datos denominada DIFUSION, la cual permite la recepción simultánea desde todas las estaciones de la red



¡!

La red permite la transferencia de programas y datos, así como mensajes y de un ordenador a otro.

*

La red dispone de un *buffer* intermedio de 256 bytes.

*

Para forzar el envío de mensajes debemos acompañar a la sentencia **PRINT #** de un **CLOSE #**.

*

La estación receptora de los datos puede recibir estos de dos formas a través de **INPUT #** o de **INKEY$ #**.

¡!

Existe una forma de envio denominada «difusion» que permite la recepcion simultánea desde todas las estaciones de la red especificada como **"n",0**

*

El interface RS232 permite enviar y recibir datos al conectarlo a cualquier periferico de serie compatible como una impresora otro ordenador o un modem de comunicaciones

*

Las transmisiones via RS232 se realizan a traves de dos canales diferentes el **"t"** para el envio de datos y el **"b"** para los codigos de control

de c representa el número de corriente asignado al canal **"t"**

El canal "t" supone codigos de 7 bits, por lo que si alguno tuviera 8, el ultimo seria ignorado

EL CANAL "b"

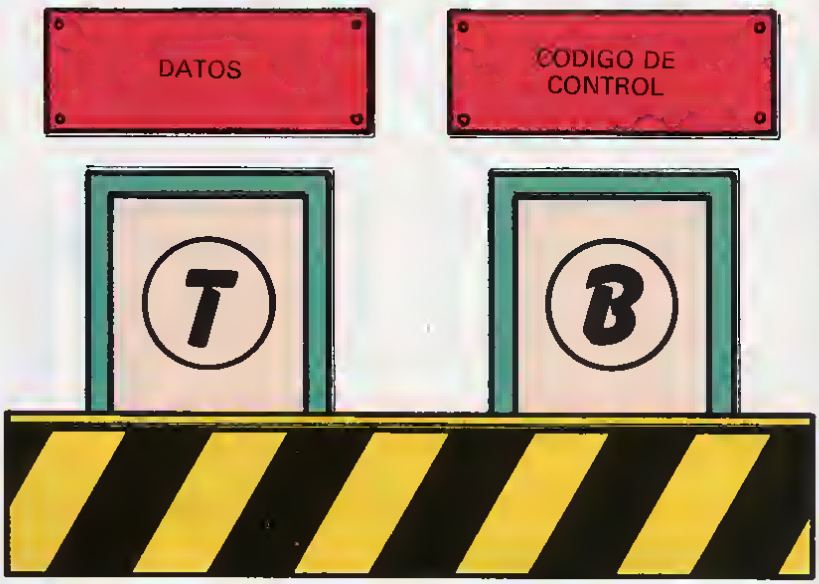
Este canal es el empleado para enviar los códigos de control a la impresora y efectuar las operaciones de transmisión de programas desde y hasta otros periféricos
Al igual que en el caso anterior, es preciso introducir la velocidad de transmisión antes de hacer uso de este canal, de la forma

FORMAT "b";v

A través de este canal es posible la comunicacion de dos Spectrum a través de la línea telefónica con el empleo de un modem Para ello, basta con cargar un programa en memoria y escribir **SAVE *"b"**, mientras en el otro lado de la linea se hace lo propio con **LOAO *"b"**. Del mismo modo, pueden enviarse bloques de memoria, pantallas, etc
Otra de las utilidades de este canal es la de enviar códigos de control a una impresora Estos varian según el modelo, permitiendo por lo general el empleo de varios tipos de letra, el uso de caracteres comprimidos y ensanchados, definir tabulaciones horizontales y verticales, etc
El uso más generalizado es trabajar con los dos canales, enviando el texto normal por el **"t"** y los controles por el **"b"**. Por ejemplo, para listar un programa en caracteres comprimidos (140 columnas en el espacio de 80) podemos escribir

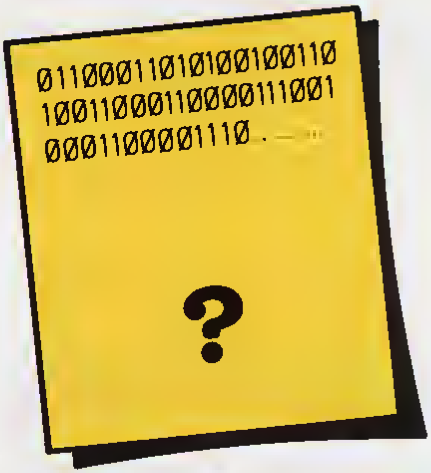
```
10 OPEN )4,"b":OPEN )5,"t"
20 PRINT )4;CHR$ 15;
30 LIST )5
40 CLOSE )5 CLOSE )4
```

Las transmisiones se realizarán a través de dos canales diferentes "t" para envio de datos y "b" para códigos de control



LA JUGADA MAS VELOZ

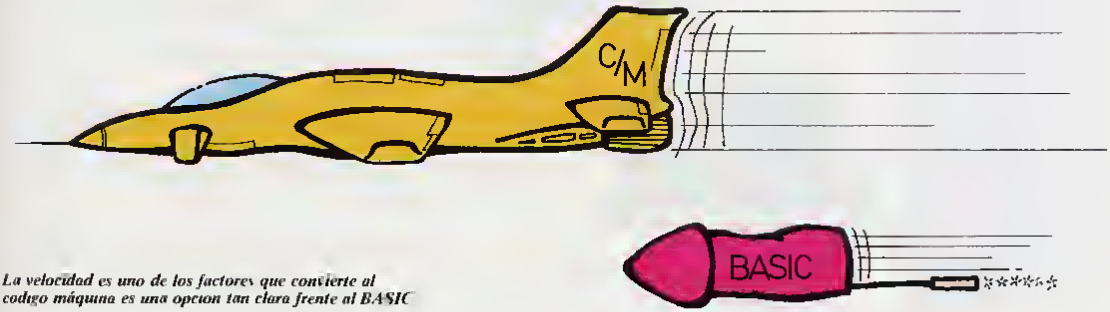
EN los capítulos anteriormente dedicados al código máquina (C/M), hacíamos una introducción a los conceptos básicos necesarios para comenzar a adentrarnos dentro de esta nueva área de la programación. Hablamos de la función **USR**, la cual nos permitía tener acceso a las rutinas escritas en este lenguaje, de reubicabilidad, concepto íntimamente ligado con el C/M, del *Firmware* o programa soporte de todas las actividades de nuestro Spectrum, de registros o pseudovariables asociadas al lenguaje máquina, etc.
Si tienes alguna duda sobre todo lo anterior, vuelve a releer con atención esos artículos, pues los conocimientos previos allí expuestos constituyen la base sobre la que vamos a partir, y sin ellos quedará dificultada la comprensión de todos los conceptos siguientes.
No estaría tampoco de más, revisar el capítulo dedicado a los sistemas de numeración, pues como veremos, se hablará de codificación y representación de instrucciones y programas, indistintamente en binario, decimal y hexadecimal, según que la claridad o el proceso lo exija.



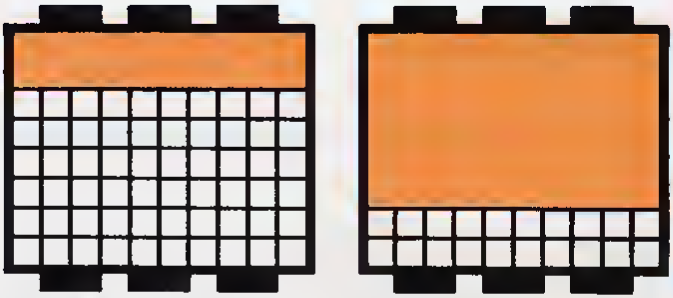
El código máquina es más difícilmente interpretable que el BASIC

LOS PODERES DEL C/M

Muchos quizá nos preguntemos qué es lo que convierte al código máquina en una herramienta tan extendida frente a un lenguaje tan popular como el BASIC. Pues bien, entre otros factores existe uno sobresaliente: la velocidad.
Ya sabemos que tarde o temprano, las instrucciones codificadas en BASIC deben traducirse de manera que el microprocesador de nuestro ordenador sea capaz de comprenderlas. Evidentemente, este proceso necesita tiempo, y es precisamente éste el que ganamos cuando realizamos nuestros programas en C/M.
Veamos un ejemplo: el siguiente programa escrito en BASIC, vuelca los primeros 6 Kbytes de me-



La velocidad es uno de los factores que convierte al código máquina en una opción tan clara frente al BASIC



Una de las ventajas del código máquina frente al BASIC es la menor ocupación de memoria.

moria ROM en la pantalla hasta dejarla, como observaremos al ejecutarlo, completamente impresentable

```
10 FOR I=16364 TO 22527
20 POKE I, PEEK (I-16384)
30 NEXT I
40 PAUSE 0
```

Una vez almacenado en la memoria, grabémoslo antes de pulsar RUN para ponerlo en marcha, y acomodémonos durante algo más de un minuto hasta que complete totalmente el trabajo. La instrucción 40 tiene como objetivo evitar que se borren las dos últimas líneas de la pantalla. Basta, pulsar al finalizar cualquier tecla, para que en su lugar aparezca el habitual mensaje **OK.** En éste y en otros programas similares es donde se pone en evidencia el BASIC demasiado lento para una sola pantalla, ¿verdad?

Sobre el código máquina se cuentan verdaderas hazañas en cuanto a su velocidad de proceso. ¿Pero este aumento es lo suficientemente apreciable, como para justificar su complejidad de manejo frente al BASIC? No adelantemos acontecimientos y comprobémoslo por nosotros mismos. Tecleemos ahora el siguiente programa

```
 5 CLEAR 30000
10 FOR I=30001 TO 30012
20 READ A: POKE I,A
30 NEXT I
40 DATA 33,0,0,17,0,64,1,0,24,237,176,
201
```

Las instrucciones anteriores sirven tan solo como cargador del código máquina impreso en la sentencia **DATA**, en la memoria RAM. Hemos escogido la dirección 30001 como origen de nuestra rutina, aunque podría haber sido cualquier otro dentro de la zona libre disponible para el usuario. De hecho, sería posible introducir los códigos uno a uno en comandos directos comenzando por **POKE 30001,33** y finalizando con **POKE 30012,201.** No te preocupes si no comprendes todavía el significado de los códigos de la línea 40; en los sucesivos capítulos serás capaz de identificarlos.

Para cerciorarte que el cargador BASIC no se encuentra en la memoria, tras ejecutar el programa anterior teclea **NEW** y luego **LIST.** La pantalla está totalmente en blanco, a excepción del mensaje **OK.** ¿no es así?

Ahora tan solo basta comprobar la presunta eficacia de nuestro programa en código máquina. Vamos a ponerlo en marcha, y para ello tecleemos **RANOOMIZE USR 30001: PAUSE 0.** Nuevamente, la orden **PAUSE 0,** tan solo sirve para que también aparezcan las dos líneas inferiores de la pantalla. Pulsa **ENTER:** espectacular, ¿no es cierto?

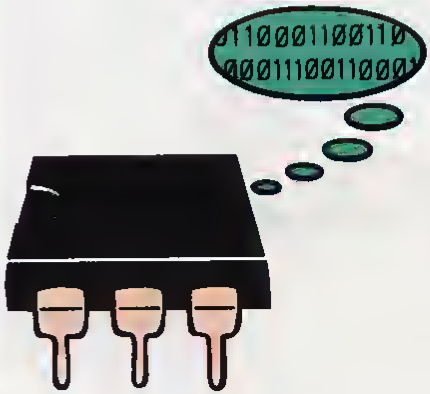
Con ello queda contrastada de forma evidente, la gran diferencia de velocidad en el proceso entre BASIC y C/M. Pero no es esta su única ventaja además, en cuanto realicemos programas cada vez más largos, echaremos en falta los Bytes de memoria que el BASIC ocupa abusivamente. Comprobémoslo.

Borra completamente la memoria ejecutando **RUN USR 0,** y vuelve a cargar el primer programa. Tras esto, introduce el siguiente comando directo, el cual nos dará la longitud del programa BASIC

```
PRINT (PEEK 23627+256*PEEK
23628)-(PEEK 23635+256*PEEK 23636)
```

la respuesta será 75, es decir, nuestro programa

El manejo del código máquina nos hará comprender la forma en que actúa la CPU de nuestro Spectrum.



¡!

Los programas confeccionados en código máquina para un ordenador, no son compatibles con otro que implemente un micro procesador diferente, y en ocasiones, ni tan siquiera estando dotados de la misma CPU

*

Con el manejo del código maquina se alcanza un conocimiento bastante profundo del sistema de funcionamiento de la CPU

ocupa 75 bytes de memoria. Sin embargo, el código máquina que realiza similar cometido, utiliza tan solo ¡12 bytes!, casi siete veces menos.

NO TODO SON VENTAJAS

Los resultados anteriores parecen indicar que el C/M desplazará al BASIC. Desde luego, esto no es cierto, y todo dependerá del tipo de aplicación que deseemos implementar. Así pues, echando un vistazo a nuestro anterior programa en código máquina, observamos que a pesar de ser más corto, al tratarse de una serie de números, es más difícil de interpretar que su equivalente en BASIC.

Además, las modificaciones y la depuración de errores pueden «volver loco» al más avezado de los programadores. Es más, todos conocemos la facilidad con la que cometemos errores de programación desde el BASIC, pero afortunadamente, nuestro ordenador, cuando no sabe por dónde salir, emite el correspondiente mensaje de error, localizable con relativa celeridad.

En cambio, el más mínimo fallo en un programa en código máquina, provoca una pequeña catástrofe, pues generalmente perdemos el control sobre nuestro Spectrum y no hay otro remedio que pulsar RESET o desconectarlo de la alimentación, para poder recuperarlo.

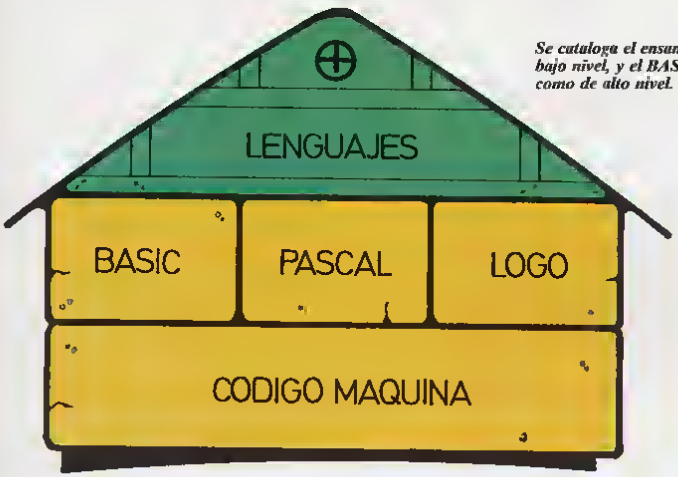
Por otra parte, el juego de instrucciones del C/M de cada ordenador es específico del microprocesador de éste (el Z 80 A en el caso de nuestro Spectrum), por lo cual los programas escritos en este lenguaje no son compatibles con otros microordenadores, mientras que desde el BASIC, normalmente podemos efectuar las modificaciones oportunas para adaptarlo.

Los cálculos matemáticos se complican en exceso, y el número de instrucciones a codificar hasta completar el mismo trabajo que en un programa BASIC se dispara, pues ha de estar todo previsto y no puede quedar libre ninguna posible fuente de error.

Todos estos inconvenientes juntos, pueden hacer

¡ERROR!

Un inconveniente del código máquina es que el mas mínimo error provoca una pequeña catastrofe: perdemos el control de nuestro Spectrum.



Se cataloga el ensamblador como lenguaje de bajo nivel, y el BASIC, PASCAL y LOGO como de alto nivel.

¡!

Los programas en sambladores proporcionan una considerable ayuda a la programacion en codigo maquina, aunque durante nuestro periodo de aprendizaje basico no son del todo recomendables.

*

Durante el proceso de traduccion de un lenguaje a otro de más bajo nivel por ejemplo. BASIC o ensamblador a codigo maquina, se genera un mayor numero de instrucciones. este fenomeno se conoce con el nombre de DESMULTIPLICACION.

¡!

El ensamblador es un lenguaje practicamente idéntico al código máquina, en el cual las instrucciones numericas (codigos) se representan por abreviaturas, mas faciles de recordar (nemonicos)

desistir a cualquiera, antes de adentrarse en el C/M. Puede ser frustrante y duro al principio, pero su manejo nos hará comprender la forma en que actúa la CPU de nuestro Spectrum y, por tanto, abrirá campos totalmente prohibidos para la programacion en BASIC

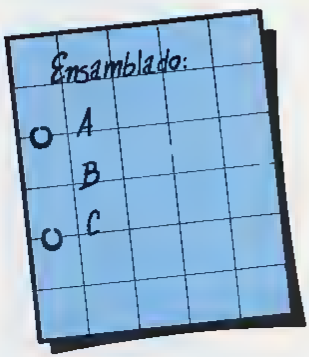
¿QUE ES UN ENSAMBLADOR?

Los programas en código máquina están construidos en el lenguaje más próximo que el microprocesador de nuestro Spectrum es capaz de reconocer, es decir, son combinaciones de unos o ceros. Analicemos la siguiente secuencia de instrucciones formada por cuatro bytes

```
00111110
00000001
00111100
01110110
```

El Z 80 A se sentirá realmente satisfecho, cuando reciba una serie de instrucciones de este tipo, pero a nosotros lo más probable es que nos deje perplejos, si afirmamos que dichos cuatro bytes son la traducción al código máquina de un programa BASIC como el siguiente:

```
LET A=1
LET A=A+1
STOP
```



Es recomendable realizar durante nuestro aprendizaje el ensamblado manual de nuestros primeros programas, y no recurrir a los ensambladores.

Los juegos de instrucciones de cada microprocesador son diferentes.

Bien, este programa es corto, pero imaginemos lo que sucedería al intentar interpretar un programa escrito totalmente en binario; sería para volverse loco, ¿verdad?

Para facilitarnos la labor, la firma ZILOG, fabricante del Z 80 A, sugirió unos «nemónicos» o códigos de operación capaces de mediar entre el ordenador y su programador, de forma tal, que a través de ellos fuera posible, de la misma manera que en BASIC usamos palabras claves como **LET** o **PRINT**, ayudar en los trabajos de programación en C/M, pero con una diferencia sustancial: una instrucción en ensamblador equivale a una instrucción en lenguaje máquina.

Por ello se cataloga al ensamblador, lenguaje de bajo nivel, a diferencia de, por ejemplo, el BASIC, PASCAL o LOGO, que son de alto nivel, pues cada instrucción escrita en uno de estos lenguajes se convierte en varias de código máquina.

Nuestro anterior miniprograma en ensamblador tendría el siguiente aspecto:

```
DIRECCION
Dir      LD A,1      ;cargar el acumu-
                     ;lador con 1
Dir+2    INC A       ;incrementar en
                     ;una unidad el contenido del acu
                     ;mulador
Dir+3    HALT        ;detenerse en este
                     ;punto
```

Los comentarios posteriores a los puntos y coma, son simplemente aclaraciones al cometido que efectúa cada instrucción, y Dir, una dirección de memoria cualquiera dentro de la RAM.

Existen en el mercado muy buenos programas que aceptan estos nemónicos (se denominan también ensambladores), aunque es recomendable que antes de trabajar con uno de ellos, realicemos nosotros mismos el ensamblado a mano, con objeto de asimilar perfectamente cual es el funcionamiento de los programas escritos en código máquina.

LONGEVIDAD

SEGÚN cuenta la leyenda, los cíclopes tenían la facultad de saber el día de su muerte nada más nacer, no queremos pecar de macabros, pero con este sencillo programa podemos asemejarnos a esos seres mitológicos y conocer, mediante una serie de preguntas-test, lo que podemos vivir en función a nuestros vicios y comportamiento: tabaco, alcohol, etc.

A más de uno, la respuesta de nuestro Spectrum le dará un susto de «muerte», o en el peor de los casos se «quedará en el sitio», al ver en la pantalla la poca cantidad de años que le quedan por disfrutar.

Pero no hay por qué preocuparse, el Spectrum es sólo un ordenador, y como tal, sigue una serie de pautas y cálculos prefijados por un programador que puede haberse equivocado en el desarrollo del programa. Aunque siempre queda la duda. En todo caso, la ejecución de este programa proporcionará sin duda un rato entretenido al realizar el test entre varios amigos. Por otra parte, es muy destacable la gran calidad estética del mismo, que indudablemente contribuye a hacernos muy agradable la ejecución de este programa, en principio bastante simple.

Parte de las preguntas que nos hace el Spectrum deben ser contestadas con "S" o "SI" y "N" o "NO" seguidos de ENTER.

EL PROGRAMA

Como acabamos de decir, el presente programa posee una gran vistosidad y facilidad de lectura, debido a la introducción y uso de la subrutina de caracteres gigantes de PSION COMPUTERS.

Dada la gran cantidad de coordenadas y literales utilizados para aumentar la belleza plástica del programa, su listado es de una longitud ciertamente considerable, si lo comparamos con la misión que lleva a cabo, pero creemos que merece la pena ocupar algo más de memoria (siempre que nos sobre) si con ello logramos mejorar la presentación del programa.

Una vez introducido el soporte BASIC, aconsejamos grabarlo en algún soporte de almacenamiento, ya sea casete, microdrive o disco, porque al existir una llamada a la subrutina en C/M (sbr caracteres gigantes) perderemos el control del ordendor si por error lo ejecutamos sin la presencia de dicha rutina.

En lo referente a la introducción de la subrutina

El programa solo analiza los casos de las personas cuyas edades estén comprendidas entre 5 y 90 años.

VAS A VIVIR
30 AÑOS

de caracteres gigantes, ya en el programa «LA BOMBA», aparecido al comienzo de nuestra obra, se hizo la descripción de las instrucciones pertinentes, debido a lo cual no redundaremos en el tema

La totalidad de las preguntas que hace el Spectrum durante el programa, pueden ser contestadas pulsando "S" (SI) o "N" (NO) seguidas de **ENTER** o bien, si aparecen en pantalla diferentes opciones, pulsando el número de la opción deseada, seguida también de **ENTER**. En el caso de que pulsemos una tecla equivocada, el Spectrum nos lo hará saber, y repetirá de nuevo la pregunta

Respecto al soporte BASIC, cuerpo fundamental de este programa, su introducción se lleva a cabo de la manera habitual, así como su grabación: **SAVE "LONG"** o **SAVE "LONG" LINE 2310**, si se desea que el programa se ejecute automáticamente al finalizar su carga

Aunque en el programa no se han empleado U.D.G. (gráficos definidos por el usuario), sí se utilizan los caracteres gráficos predefinidos por Sinclair, que se obtienen mediante la pulsación de los números comprendidos entre 1 y 8 en el modo **GRAPHICS**

En el listado, los gráficos simples, es decir, los que se consiguen mediante la pulsación de la tecla numérica necesaria, se representan por el número que designa la tecla con un subrayado simple

Los gráficos cambiados son los que se obtienen pulsando la tecla numérica en cuestión al tiempo que **CAPS SHIFT** (siempre en modo **GRAPHICS**), y se representan de forma similar a los gráficos simples anteriormente citados, pero con un subrayado doble. Recordemos que la entrada y la salida al modo **GRAPHICS** se consigue mediante la pulsación simultánea de **CAPS SHIFT** y **9**, lo cual producirá el efecto de cambio a cursor **G**.

¡Suerte, y a vivir muchos años!

```
 10 REM ************************
 20 REM * J.M MAYORAL SERRANO *
 30 REM ************************
 40 REM * TEST VIDA  (C) 1985 *
 50 REM ************************
 60 POKE 23658,8
 70 LET PA=7
 80 BORDER 1: PAPER 1: CLS
 90 LET K=540
100 LET C=0
110 LET CO=0
120 LET P$="#  TEST DE LONGEVIDAD  #"
130 BRIGHT 1: INK 7
140 LET YY=16
150 LET XS=1: LET YS=3
160 PRINT  PAPER 6; INK 2;AT 0,0;"433333333333333333333333333333337"
170 PRINT  PAPER 6; INK 2;AT 6,0;"133333333333333333333333333333332"
180 FOR N=1 TO 5
190 PRINT  PAPER 6; INK 2;AT N,0;"5";AT N,31;"5"
200 NEXT N
205 BRIGHT 0
210 GO SUB 2170: GO SUB 2410
215 BRIGHT 1
220 PRINT  PAPER 6; INK 2;AT 8,0;"433333333333333333333333333333337"
230 FOR N=9 TO 19
240 PRINT  PAPER 6; INK 2;AT N,0;"5";AT N,31;"5"
250 NEXT N
260 PRINT  PAPER 6; INK 2;AT 20,0;"133333333333333333333333333333332"
270 GO SUB 2340
280 LET A=71
290 LET P$="QUE EDAD TIENES ?"
300 LET YY=105
310 LET YS=2
320 PAPER 7: INK 2
340 GO SUB 2170
350 INPUT B
355 IF B<90 OR B>5 THEN  GO TO 350
360 GO SUB 2340
370 LET P$="  ESTAS CASADO/A ?  (S/N)  "
380 INK 1
390 PAPER 6
400 GO SUB 2170
410 INPUT A$
420 IF A$="S" THEN  LET A=76: GO TO 460
430 IF A$="N" THEN  GO TO 460
440 GO SUB 2340: GO SUB 2500
450 GO TO 370
460 GO SUB 2340
470 LET P$="TE CONSIDERAS"
480 LET YY=75
490 BRIGHT 0
500 PAPER 7: INK 2
510 LET XS=2: LET YS=2: GO SUB 2170
520 LET P$="1)   RICO"
530 LET XS=1
540 LET XX=40: LET YY=100
550 INK 1
560 GO SUB 2190
570 LET P$="2)   POBRE"
580 LET YY=120
590 INK 4
600 GO SUB 2190
610 LET P$="3)   NI LO UNO NI LO OTRO"
620 LET YY=140
630 INK 5
640 GO SUB 2190
650 INPUT C$
660 IF CODE C$<49 OR CODE C$>51 THEN  GO TO 700
670 LET VAL=VAL C$
680 LET A=A+4-VAL
690 GO TO 730
700 GO SUB 2340
710 GO SUB 2500
720 GO TO 470
730 GO SUB 2340
740 LET P$="  TE SOBRAN KILOS ? (S/N)  "
750 LET YY=105
760 LET XS=1
770 PAPER 3: INK 7: BRIGHT 1
780 GO SUB 2170
790 INPUT D$
800 IF D$="S" THEN  GO TO 850
810 IF D$="N" THEN  GO TO 850
820 GO SUB 2340
830 GO SUB 2500
840 GO TO 740
850 GO SUB 2340
860 IF D$="N" OR B<45 THEN  GO TO 960
870 LET P$="CUANTOS ?"
880 LET YY=100
890 LET YS=4: LET XS=3
900 FLASH 1
910 PAPER 2: INK 6
920 GO SUB 2170
930 INPUT P
940 LET P=INT (P*1.27): LET C=C+P
950 FLASH 0: GO SUB 2340
960 LET YY=75
970 INK 2: PAPER 7: BRIGHT 0
980 LET XS=1: LET YS=3
990 LET P$="HACES EJERCICIO ?"
1000 GO SUB 2170: LET YY=100
1010 LET YS=2: BRIGHT 0: INK 1
1020 LET P$="1)   NADA"
1030 GO SUB 2190
1040 LET P$="2)   ALGO"
1050 LET YY=120
1060 INK 4
1070 GO SUB 2190
1080 LET P$="3)   BASTANTE"
1090 LET YY=140
1100 INK 3
1110 GO SUB 2190
1120 INPUT E
1130 IF E>3 OR E<1 THEN  GO SUB 2340: GO SUB 2500: GO TO 960
1140 LET A=A-1+2*E
1150 GO SUB 2340
1160 LET P$=" ERES NERVIOSO/A ? (S/N) "
1170 LET YY=105
1180 BRIGHT 1
1190 INK 0: PAPER 5
1200 GO SUB 2170
1210 INPUT F$
1220 IF F$="S" THEN  LET A=A-6: GO TO 1260
1230 IF F$="N" THEN  LET A=A+6: GO TO 1260
1240 GO SUB 2340: GO SUB 2500
1250 GO TO 1160
1260 GO SUB 2340
1270 LET P$="BEBES ALCOHOL?"
1280 LET YY=75
1290 LET XS=2
1300 BRIGHT 0: PAPER 7
1310 INK 2
1320 GO SUB 2170
1330 LET YY=100: LET XX=40
1340 LET XS=1
1350 INK 1
1360 LET P$="1)   NADA"
1370 GO SUB 2190
1380 LET P$="2)   ALGO"
1390 INK 4: LET YY=120
1400 GO SUB 2190
1410 LET P$="3)   MUCHO"
1420 LET YY=140: INK 2
1430 GO SUB 2190
1440 INPUT G
1450 IF G<1 OR G>3 THEN  GO SUB 2340: GO SUB 2500: GO TO 1270
1460 LET G=5*(G-1)
1470 IF B>A THEN  LET A=B+(B-A)/2
1480 GO SUB 2340
1490 LET P$="  FUMAS ?    (S/N)"
1500 BRIGHT 1
1510 LET YY=105
1520 PAPER 1: INK 7
```

```
1530 GO SUB 2170
1540 INPUT H$
1550 GO SUB 2340
1560 IF H$='S' THEN  LET A=A-5: GO TO 1590
1570 IF H$='N' THEN  GO TO 1590
1580 GO SUB 2500: GO TO 1490
1590 LET P$='ENFERMAS FACILMENTE ?  (S/N)'
1600 INK 1: PAPER 4
1610 GO SUB 2170
1620 INPUT K$
1630 IF K$='S' THEN  LET A=A-6: GO TO 1670
1640 IF K$='N' THEN  LET A=A+7: GO TO 1670
1650 GO SUB 2340: GO SUB 2500
1660 GO TO 1590
1670 GO SUB 2340
1680 LET P$='ERES HOMBRE O MUJER ?'
1690 LET YY=75
1700 PAPER 7: BRIGHT 0
1710 INK 2
1720 LET YS=3
1730 GO SUB 2170
1740 LET YS=2: LET YY=110
1750 LET XX=45
1760 LET P$='1)   HOMBRE': INK 1
1770 GO SUB 2190
1780 LET YY=130
1790 INK 4
1800 LET P$='2)   MUJER'
1810 GO SUB 2190
1820 INPUT S$
1830 IF S$='1' THEN  LET S$='H': GO TO 1870
1840 IF S$='2' THEN  LET S$='M': GO TO 1870
1850 GO SUB 2340: GO SUB 2500
1860 GO TO 1680
1870 LET PA=2
1880 GO SUB 2340
1890 LET YS=3: LET XS=1: INK 7
1900 LET YY=80
1910 BRIGHT 1: PAPER 2
1920 LET P$=' #  PREVISION  DE  VIDA  # '
1930 GO SUB 2170
1940 BRIGHT 0
1950 INK 6
1960 LET XX=90: LET YY=110
1970 IF S$='M' THEN  GO TO 2030
1980 LET AN=INT (A-C/5-G)
1985 GO SUB 5000
1990 LET P$=STR$ AN
2000 GO SUB 2190
2005 LET XX=125: LET P$='anos'
2007 GO SUB 2190
2010 GO SUB 2830
2020 GO TO 2070
2030 LET AN=INT (A+7-C/5-G)
2035 GO SUB 5000
2040 LET P$=STR$ AN
2050 GO SUB 2190
2055 LET XX=125: LET P$='anos'
2057 GO SUB 2190
2060 GO SUB 2830: PAUSE 100
2070 LET P$='DESEAS HACER OTRO TEST ?'
2080 INK 7
2090 LET YY=140
2100 LET YS=2: LET XS=1
2110 GO SUB 2170
2120 GO SUB 2750
2130 LET K$=INKEY$
2140 IF K$='S' THEN  LET PA=7: GO SUB 5050: GO TO 270
2150 IF K$='N' THEN  GO TO 10000
2160 GO TO 2120
2170 REM SBR. CARACT. GIGANTES
2180 LET xx=(256-8*xs*LEN p$)/2
2190 LET i=23306
2200 POKE i,xx: POKE i+1,yy: POKE i+2,xs: POKE i+3,ys
: POKE i+4,8
2210 LET i=i+4: LET w=LEN p$
2220 FOR q=1 TO w
2230 POKE i+q,CODE p$(q)
2240 NEXT q
2250 POKE i+w+1,255
2260 BEEP .1,40
2270 LET w=USR 32256
2280 BEEP .05,40: BEEP .05,40
2290 RETURN
2300 REM  AUTO-EJEC. OBLIGATORIA
2310 CLEAR 32255
2320 LOAD 'LIT'CODE : RUN
2330 REM  A BLANCO
2340 FOR W=9 TO 19
2350 BEEP .005,40
2360 PRINT  PAPER PA; BRIGHT 0;AT W,1;'

2370 NEXT W
2380 FLASH 0
2390 RETURN
2400 REM  OVER EN TITULO
2410 OVER 1
2420 BRIGHT 1
2430 FOR W=2 TO 5
2440 PRINT  INK W+2;AT W,5;'
2450 NEXT W
2460 OVER 0
2470 BRIGHT 0
2480 RETURN
2490 REM  MENSAJE
2500 LET X1=XS: LET Y1=YS
2510 LET Y2=YY: LET X2=XX
2520 LET T=INT (RND*8)
2530 LET P=INT (PND*7)+1
2540 IF T=P THEN  GO TO 2530
2550 FOR W=9 TO 19
2560 PRINT  PAPER P; INK T;AT W,1;'6666666666666666666666666666'
2570 NEXT W
2580 LET P$='REPITO !!'
2590 LET YY=85
2600 LET XS=3: LET YS=8
2610 LET T=INT (RND*8)
2620 LET P=INT (RND*6)+2
2630 IF P=T THEN  GO TO 2620
2640 INK T
2650 PAPER P
2660 GO SUB 2170
2670 FOR W=1 TO 10
2680 BEEP .05,40
2690 NEXT W
2700 GO SUB 2340
2710 LET XS=X1: LET YS=Y1
2720 LET YY=Y2: LET XX=X2
2730 RETURN
2740 REM  COLOR MENSAJE
2750 IF CO>7 THEN  LET CO=0
2760 IF CO=2 THEN  LET CO=3
2770 PRINT  OVER 1; INK CO;AT 17,2;'

2780 PRINT  OVER 1; INK CO;AT 18,2;'

2790 PRINT  OVER 1; INK CO;AT 19,2;'

2800 LET CO=CO+1
2810 RETURN
2820 REM  ADAPT. LETRA
2830 PLOT 134,63
2840 DRAW 3,0
2850 PLOT 134,62
2860 DRAW 3,0
2870 RETURN
4999 REM  VERIF. ANNOS
5000 IF AN>B THEN  RETURN
5010 LET ANN=INT ((B-AN)/5)
5020 LET AN=B+ANN
5030 RETURN
5040 REM  COMIENZO
5050 LET K=540
5060 LET C=0
5070 LET CO=0
5080 BRIGHT 1
5090 RETURN
```